Tad: Generalized Transformer-Based Domain-Aware Hate Speech Detection

Ruijie Xi, Munindar P. Singh

North Carolina State University
Raleigh, North Carolina 27606 USA
rxi@ncsu.edu, mpsingh@ncsu.edu

Abstract

Warning: This paper contains examples of hateful content. Please be aware that this content could be offensive.

Hate speech targets different social groups such as race and gender, and poses a significant threat to social harmony. Researchers are increasingly motivated to devise efficient techniques to improve automatic hate speech detection on social media platforms. However, current models are usually evaluated without considering hate speech targets and fail when the targets are unseen in the training data.

In this study, we examine target (domain) shifts of hate speech and propose *Tad*, an adaptation framework for neural models that adopts domain-aware networks to improve cross-domain hate speech detection. *Tad* features a hate knowledge lexicon infusion network, a domain-specific network, and a weighting network. We demonstrate that incorporating *Tad* improves the performance of leading neural models in hate speech detection when tested on unseen domains. Specifically, *Tad* yields improvements of up to 8.1% and an average of 2.4% in macro F1-scores. Moreover, we identify data quality and quantity as vital factors to address performance gaps between models tested on seen and unseen domains. Our results reveal that excessive knowledge infusion may result in a decrease in performance such as for *Religion*. In addition, we find trade-offs in cross-domain hate speech detection. For example, weighted loss for heavily imbalanced data generally improves performance.

1 Introduction

The openness and reach of social media are exploited by malefactors to engage in hate speech targeting individuals or groups based on characteristics such as race, gender, or ethnicity. Hate speech can lead to cyber and offline crimes against minorities (Mathew et al. 2019). Automatic hate speech detection is a way to protect vulnerable users and promote inclusivity in online platforms. Consequently, researchers are increasingly motivated to develop effective methods for detecting hate speech on social media platforms (Waseem and Hovy 2016; ElSherief et al. 2018; Arango, Pérez, and Poblete 2019; Ousidhoum et al. 2019; Chiril et al. 2022; Mathew et al. 2021; Maity et al. 2022).

Hate speech detection in online communication is an important application of Natural Language Processing (NLP). Previous models of hate speech detection (Waseem and Hovy 2016; Vigna et al. 2017; Swamy, Jamatia, and Gambäck 2019; Mossie and Wang 2020; Kamal et al. 2023) perform well when combining all training data from certain benchmark datasets related to specific *target groups* (e.g., race). These approaches require a large number of hate speech instances to achieve high performance. However, real-world applications necessitate detecting hateful content in social media when targets are new in training data. This is due to targets constantly emerging. We can identify "domains" in hate speech based on target groups (Ludwig et al. 2022) and corpora (Chiril et al. 2022; Sarwar and Murdock 2022). We focus on **target groups** to align with real-world applications.

Hateful language varies across different domains, targeting groups like immigrants and women (Ludwig et al. 2022; Sarwar and Murdock 2022). However, classifiers for hate speech struggle to adapt to new domains (Bose, Illina, and Fohr 2021; Fortuna, Soler-Company, and Wanner 2021), losing up to 50% of their performance scores in *out-domain* (where train and test data are from different targets) vis-á-vis *in-domain* (where train and test data share the same target) experiments (Arango, Pérez, and Poblete 2019). This challenge is exacerbated by domain shifts, where seemingly innocent language contains offensive terms within specific communities, such as "*ni**a*" when used by African American community (Vigna et al. 2017). Annotating hateful content for new domains is labor-intensive, complicated by biases in data collection and disagreements in annotation (Dixon et al. 2018; Davani, Díaz, and Prabhakaran 2022). Therefore, constructing robust hate speech detection classifiers faces challenges when there's a disparity between training and testing domains.

Another drawback is using a shared, domain-agnostic input representation for different domains, limiting domain-specific knowledge obtained during training (Liu, Zhang, and Liu 2018). This approach may overlook domain nuances, impacting task performance. Addressing this challenge requires techniques to effectively integrate domain-specific information, enhancing model adaptability and performance across diverse domains. For instance, the word "*bi**h*" may indicate hate speech targeting women in one domain, but it is commonly used in rap lyrics without derogatory connotations.

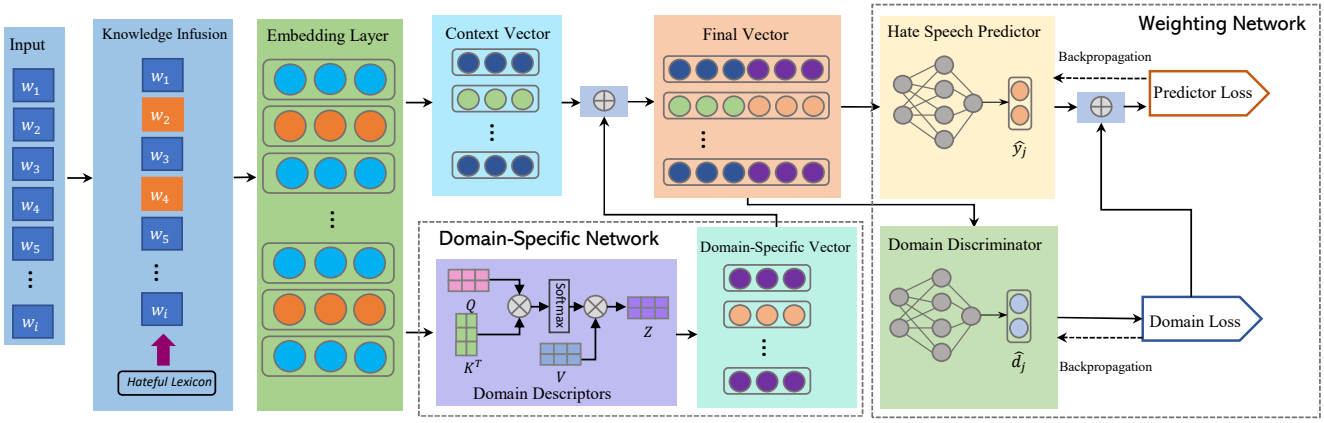


Figure 1: Framework of Tad. Here, w_i represents a token in an instance, $\widehat{y_j}$ is the predicted hate label, and $\widehat{d_j}$ is the prediction of whether an instance belongs to training domains (i.e., $\widehat{d_j} \in [0, 1]$). Hateful tokens in ■ are detected by an additional lexicon.

We study how performance varies across hate speech targets and propose a neural adaptation framework to model domain shifts. We examine HateXplain (Mathew et al. 2021), a dataset that annotates each instance with hate and target group labels. HateXplain labels nonhateful instances unlike other multidomain datasets (ElSherief et al. 2018; Ousidhoum et al. 2019; Basile et al. 2019), which enables us to select diverse examples that encompass various domains. This enables us to investigate the effectiveness of our approaches thoroughly. We consider three domains: *Ethnicity* (e.g., Asian), *Religion* (e.g., Buddhism), and *Sex Orientation (sexorient)* (e.g., Asexual).

Methods We compare cross-domain hateful language distributions and analyze target shift impacts on hate speech detection via cross-domain experiments, training regression classifiers on one domain and evaluating on another. Statistical quantification of domain shifts using SHAP values highlights the importance of adapting from seen to unseen data. To bridge performance gaps, we introduce *Tad*, a neural framework featuring *knowledge infusion* leveraging a toxicity lexicon (Bassignana, Basile, and Patti 2018), a domain-specific network adopting attention mechanisms for domain descriptors (Vaswani et al. 2017), and a weighting network dynamically adapting to domain shifts using a domain discriminator to optimize training weights for tested domains (Jiang and Zhai 2007). Our framework is shown in Figure 1.

Contributions As shown in Table 3, incorporating Tad improves neural models' performance for hate speech classification in cross-domain scenarios, yielding an average F1-score improvement of 2.4% and up to 8.1% on *Ethnicity*. Our findings underscore the importance of learning domain-specific hateful language for effective cross-domain hate speech detection. Models leveraging external domain-specific hateful knowledge, such as *Sexorient*, demonstrate significant performance gains, averaging 2.5% when adapting to domain shifts. Furthermore, we observe that different technologies exhibit varying strengths and limitations in hate speech detection. While weighted loss generally enhances performance, it may lead to a decline in out-of-domain performance for *Sexorient*. We will release our data, code, and supplementary material if the paper is published.

2 Related Work

We discuss the literature on hate speech datasets and methods.

Datasets Researchers have collected hate speech datasets from social media (Waseem and Hovy 2016; ElSherief et al. 2018; Ousidhoum et al. 2019; Mathew et al. 2021). While some are categorized by target groups like immigrants and women (Waseem and Hovy 2016; Basile et al. 2019), most are limited to single domains. This lack of domain-specific knowledge leads to biased datasets due to biased sampling procedures and drop performance of classifiers (Arango, Pérez, and Poblete 2019). For instance, Wiegand, Ruppenhofer, and Kleinbauer (2019) demonstrate that Waseem and Hovy's (2016) dataset predominantly features hate speech instances generated by a small group of authors, with domain-specific keywords like *announcer* related to women's competence in sports frequently appearing. Despite the recent trend of annotating fine-grained targets of hate speech on social media, many previous datasets lack target labels for nonhateful instances (Toraman, Şahinuç, and Yilmaz 2022; Zampieri et al. 2023), or conflate the concepts of "hateful" and "offensive," (Almohaimeed et al. 2023) which should be distinguished (Fortuna, Soler-Company, and Wanner 2021).

Methods Supervised classifiers like Logistic Regression (LR) and Support Vector Machines (SVM) are widely used in hate speech detection (Waseem and Hovy 2016; Vigna et al. 2017; Swamy, Jamatia, and Gambäck 2019; Kamal et al. 2023). Recent approaches incorporate word embedding representations with Convolutional Neural Network (CNN) (Zhang, Robinson, and Tepper 2018; Roy, Bhawal, and

Subalalitha 2022; Ghosh et al. 2023) and Recurrent Neural Network (RNN) variants, including LSTM (Badjatiya et al. 2017; Vigna et al. 2017), Gated Recurrent Unit (GRU) (Mossie and Wang 2020), and transformer-based models (Swamy, Jamatia, and Gambäck 2019). Capsule networks (Kamal et al. 2023) are also utilized to capture textual representations. Various strategies address domain adaptation in hate speech detection. Chiril et al. (2022) demonstrate that integrating domain-specific knowledge, such as emotion and hateful lexical knowledge, improves model performance. Sarwar and Murdock (2022) propose a method to augment hate speech data through instance reconstruction. However, these approaches face limitations due to dataset incompatibilities, posing methodology challenges (Zhou et al. 2021). For instance, Swamy, Jamatia, and Gambäck (2019) find that BERT (Devlin et al. 2019) can transfer knowledge between domains, but dataset incompatibilities remain a primary obstacle. Additionally, Fortuna, Soler-Company, and Wanner (2021) stress the importance of accurate and non-overlapping definitions of hate speech across datasets. Furthermore, Ludwig et al. (2022) observe that unsupervised methods can lead to negative knowledge transfer in cross-domain hate speech detection due to incorrect pseudo-labels interfering with training.

3 Dataset

This section introduces our dataset and approaches.

3.1 Dataset

We adopt HateXplain (Mathew et al. 2021), which consists of around 20K annotated anonymized posts from Twitter and Gab. The dataset was primarily annotated with the class labels "Normal," "Offensive," and "Hate," as well as annotations of target groups such as *LGBTQ*. HateXplain distinguishes "Offensive," and "Hate," because Mathew et al. (2021) argue that many messages can be offensive without qualifying as hate speech. Unlike other datasets with target annotations (ElSherief et al. 2018; Ousidhoum et al. 2019), HateXplain includes annotations for all data points (Ludwig et al. 2022), including those in the "Normal" class. Therefore, we conduct experiments exclusively on HateXplain to evaluate our method's knowledge transfer abilities in strictly separated target groups.

Domains	Target groups	# H	# N	# O
Ethnicity	African, Asian, Hispanic, Indian, Jewish, Caucasian, Arab	2,103	880	917
Religion	Buddhism, Christian, Islam, Nonreligious, Hindu	393	307	248
Sexorient	Bisexual, Asexual, Heterosexual, Homosexual	201	422	470

Table 1: The domains and target groups in our curated dataset based on the HateXplain dataset (Mathew et al. 2021) (H: Hate, N: Normal, and O: Offensive).

Ludwig et al. (2022) extend the HateXplain dataset for exploring domain adaptation technologies, but their curated dataset is not public. Although they refer to three domains, namely *Gender*, *Religion*, and *Race*, they do not clarify the targeted communities that were initially annotated (e.g., African, Asian, and Indian). Hence, we follow Ludwig et al.'s (2022) method on categorizing the target groups by ourselves based on the initiated annotations. HateXplain employs three annotators for each instance. To ensure that the trained models generalize effectively from a single domain to a new one, we collect instances where all three annotators agree on assigning a sole target group label. We focus on three categories. *Ethnicity*, *Religion*, and *Sexorient*—because other target groups (e.g., *Disability*) contain fewer than 60 instances annotated as "Hate," risking poor results due to insufficient coverage of all class labels. Table 1 displays data distributions.

4 Analysis: Target Shift Impacts

To qualify target shifts of hate speech language and how these shifts affect hate speech classification, we conduct a classification evaluation under a cross-domain setting that trains a classifier on one domain and tests the classifier on the other domains. We split 80% of documents as the training set and hold out 20% of documents as the testing set for each domain corpus. We extract TF-IDF weighted uni-, bi-, and tri-gram features on each domain corpus with the most frequent 15,000 features during the training. We then build a logistic regression classifier using LogisticRegression from scikit-learn[1] with default parameters. Finally, we evaluate the classifier across each domain's test set using macro F1 scores.

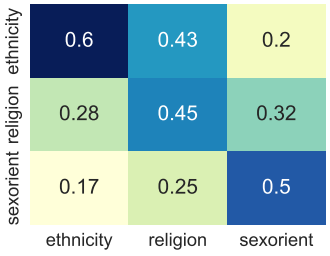
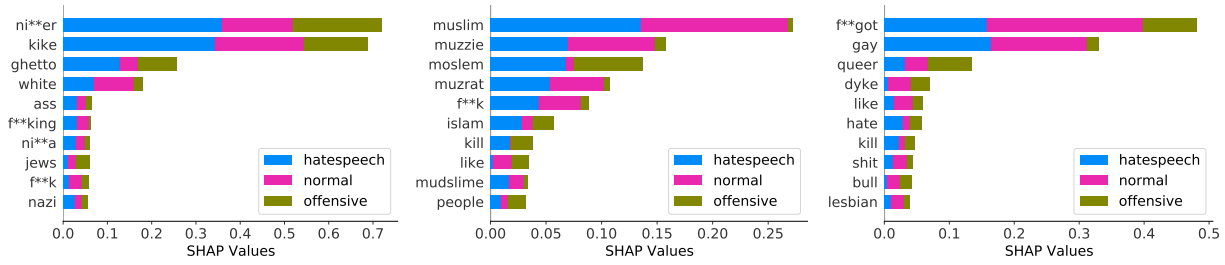


Figure 2: In-Domain prediction results for each target, the X-axis represents the training sets, whereas the Y-axis represents the testing sets used in logistic regression models.

As illustrated in Figure 2, in-domain scores exceed out-domain scores. For example, the in-domain evaluation on *Ethnicity* achieves 60.4% versus out-domain evaluation when it becomes the tested new domain, ranging from 45.1% to 50.3%. The finding applies to other evaluations. To qualitatively examine the domain shifts, we extract top predictable word features for each domain by calculating SHapley Additive exPlanations (SHAP) values (Lundberg

[1]https://scikit-learn.org/stable/



(a) *Ethnicity*. (b) *Religion*. (c) *Sexorient*.

Figure 3: Top ten predictive features computed using SHAP (Lundberg and Lee 2017). Larger values indicate greater importance for explaining predictions.

and Lee 2017) using `shap`.[2] Computing SHAP values is a method for explaining the output of machine learning models in terms of the contribution of individual features to the model's predictions. The top ten predictable features of each domain for the three labels are shown in Figure 3.

We observe that classifiers' performance vary cross domains and each domain has domain-specific language, while some words are crucial in prediction performance. For example, in *Sexorient*, the word *kill* has larger SHAP values in *Normal* compared to that in *Religion*. Our results suggest that learning domain-specific knowledge is crucial. Furthermore, dynamically adapting the learned knowledge to optimize word weights from seen to unseen domains is necessary. Therefore, we propose a neural adaptation framework to model domain shifts, comprising three modules:

Knowledge infusion adopts extra knowledge to infuse hateful content for unseen domains.

Domain-specific network learns the most salient hate content for each domain.

Weighting network adapts to domain shifts using the domain-specific knowledge learned from seen domains.

5 Methods: Learning to Adapt Framework

Table 2 describes the notation used in this section.

Problem Reformulation Given hate speech instances drawn from m distinct domains $\{D_i\}_{i=1}^{m}$ (i.e., in in-domain, $m = 3$; in out-domain, $m = 2$), where D_i contains data points that consist of (s_j, d_j, y_j) where s_j is a sequence of words represented as $w_1, w_2 \dots w_{|s_j|}$. Here, d_j is the domain label for s_j. Our objective is to find a function f that maps each instance (s_j, d_j) to its corresponding hate speech label $y_j \in [0, 1]$. Our challenge is how to improve the generalization f by identifying the correlations between domains.

Knowledge Infusion To enhance a model's capacity to identify hateful content, we infuse a lexicon during training. We adopt HurtLex (Bassignana, Basile, and Patti 2018), a popular lexicon for detecting hateful language on social media (Jiang and Zubiaga 2021; Chiril et al. 2022). HurtLex contains 6,287 offensive, aggressive, and hateful English words. We reprocess the input and generate a vector for an instance s_j as $z_1, z_2, \dots, z_{|s_j|}$, where if w_i in the lexicon, $z_i = 1$, else, $z_i = 0$. We include an additional loss $L_k = -\sum_i |c_i| z_i$, where c_i denotes BERT's output attention weights for w_i, indicating the salience of each token (Jain et al. 2020). The term L_k decreases the loss when the important tokens are hateful; it has no effect for nonhateful tokens. The additional reward emphasizes the appearance of hateful words, which benefits the model identifying hate content.

[2]https://shap.readthedocs.io/en/latest/

Notation	Description
s_j	An input instance
d_j	Domain of s_j
w_i	ith token in an input sentence s_j
D_i	Data points of ith domain
y_j	Hate speech label for an instance s_j
m	Number of all domains
h_t	Hidden states of s_j
Q	Projection matrix that linearly project h_t
M	Domain descriptor matrix
K	Dimension of M
P	Projection matrix that linearly project M
U_j	Domain-specific representation for s_j
v	A linear network to obtain U_j by projecting M and h_t
a_j	Similarity between s_j and M
z_j	Indication of whether a token belongs to the lexicon
λ	Parameters of the knowledge infusion
θ	Parameters of the prediction network
ϕ	Parameters of the weighting network

Table 2: The notation used in this paper.

Domain-Specific Network As mentioned in Section 1, domains exhibit language differences that alter the salience of hate signals. Whereas previous works use domain-agnostic representations (Chiril et al. 2022; Ludwig et al. 2022), we explicitly capture domain-specific representations using *domain descriptors* (Liu, Zhang, and Liu 2018). Given an input (s_j, d_j, y_j), we apply an embedding layer (BERT encoder) and BiLSTM to generate its general semantic representation $h_t = h_1, h_2, \dots, h_{|sj|}$. We denote domain descriptors as a matrix $M^{K \times m}$, where K is the dimension of the input representations (the same as h_t) and m indicates the number of domains. The matrix M is automatically learned. We weigh hidden states h_t by each domain descrip-

tor M_i for obtaining its domain-specific representation, U_j^i. Here, U_j^i is the weighted sum of h_t of attention scores a_{jt}^i:

$$U_j^i = \sum_{t=1}^{|sj|} a_{jt}^i h_t, i \leq m, \tag{1}$$

where a_{jt}^i sums up to one and reflects the similarity between the ith domain descriptor M_i and the hidden state h_t. To calculate a_{jt}^i, we select the additive attention mechanisms (Bahdanau, Cho, and Bengio 2015) over dot-product (Vaswani et al. 2017) because doing so yields better performance in our experiments. We adopt a one-hidden layer feed-forward network (Liu, Zhang, and Liu 2018) to calculate a_{jt}^i as:

$$\begin{aligned} {u_{jt}}^i &= v^T \tanh(PM_i + Qh_t), \\ {a_{jt}}^i &= \frac{\exp({u_{jt}}^i)}{\sum_{p=1}^{|s_j|} \exp({u_{jp}}^i)}, i \leq m, t \leq |s_j|, \end{aligned} \tag{2}$$

where P and Q are matrices that linearly project M and h_t vectors into a $2K \times K$ space. And, we normalize a_{jt}^i via softmax. Here, v serves as the linear layer after taking $\tanh$ of the sum of the two vectors, where the output dimension is set to $2K$. It is empirically beneficial to project the representation vectors to a larger space in our experiments. Furthermore, we apply a self-attention layer to model domain relations for simultaneously optimizing the ith domain descriptor, M_i. We compute dot products between M and every other domain descriptor and normalize the results using softmax—that is,

$$\begin{aligned} M' &= M \cdot \text{softmax}(M^T M_i), \\ \text{softmax}(x_i) &= \frac{\exp(x_i)}{\sum_j \exp(x_j)}, \end{aligned} \tag{3}$$

where M' is the updated descriptor for each domain.

Weighting Network To improve the generalization performance, we incorporate an instance weighting network (Jiang and Zhai 2007; Huang, Wormley, and Cohen 2022) that dynamically adapts to domain shifts of hateful language. We leverage a *discriminator* to optimize training weights for target domains. The discriminator is a binary classifier that predicts whether the input instances belong to source domains. The discriminator calculates in-domain probabilities, $f_\phi(d_j|h_t, U_j)$, where ϕ are the corresponding parameters. We use cross-entropy loss for the weighting network as:

$$L_d = L(\widehat{d_j}, f_\phi(h_t, U_j)), \widehat{d_j} \in [0, 1] \tag{4}$$

Thus, the weighting network dynamically adjusts the parameters, including those in the domain-specific network, such as a_{jt}^i. We optimize the prediction and weighting networks separately, which enhances control of the training process.

Prediction We obtain the pretrained embeddings using neural models that map sequences into dimensional embedding representations of their textual context. We concatenate the weighted domain-specific representation with the corresponding context embeddings so the model can learn both domain-specific representations and context vectors of an input instance. Then, we use a fully connected network that applies softmax to predict $\widehat{y_j} \in [0, 1]$, which indicates whether an input is hate speech. Following Mathew et al. (2021), we apply dropout on the concatenated vectors and employ a tanh activation function. We apply cross-entropy to calculate the hate speech detection loss and optimize the network by calculating $L_y = L(y_j, f_\theta(\widehat{y_j}|h_t, U_j))$, where θ denotes the set of parameters including domain descriptors, attention weights, and softmax parameters. In the out-domain scenario, since the last layer has m outputs, we adopt an *ensemble approach* (Liu, Zhang, and Liu 2018) to obtain a single output for target domains during tuning. In particular, since the predictor outputs probabilities on how likely a test instance comes from the source domain, we use these probabilities as weights to average each output predicted domain label. Our experimental findings suggest that this approach yields better performance than using the maximum probability. Combining the losses, the objective for our model becomes:

$$L = \arg\min_{\theta,\phi,\lambda} \cdot (L_y + L_d + \lambda L_k) \tag{5}$$

where λ is a coefficient for controlling the importance of knowledge infusion.

6 Experiments

Our curated dataset contains 5,741 instances from three domains. To evaluate the effectiveness of our methods in enhancing hate speech classification, we implement our methods on established deep learning models known for their strong performance in in-domain hate speech detection. By doing so, we can evaluate the efficiency of our methods in modeling domain shifts in cross-domain scenarios.

6.1 In- and Out Domain Settings

As in previous domain adaptation works (Chiril et al. 2022; Ludwig et al. 2022; Huang, Wormley, and Cohen 2022), we evaluate Tad's performance in two settings. The in-domain setting trains, validates, and tests models on the same domain. The out-domain setting trains models on two domains, tunes model parameters on the third domain, and evaluates them on the new domain. For example, we train models on the *Sexorient* and *Religion* domains, then tune and test on *Ethnicity* domain. For in-domain experiments, we randomly split 80%, 10%, and 10% of instances into training, development, and test sets. For out-domain experiments, we randomly select 80% of the instances from each domain as test sets and the remaining 20% for tuning.

6.2 Models

We employ Tad on neural models with established high in-domain hate speech detection performance as suggested by previous work (Ding, Zhou, and Zhang 2019; Mossie and Wang 2020; Sarkar et al. 2021; Caselli et al. 2021). In the in-domain experiments, we retain the default modules of the models. For out-domain experiments, we adapt the modules used in the in-domain experiments and then integrate Tad after encoding the input instances.

	Ethnicity		Religion		Sexorient	
Model	No adapt	Adapt	No adapt	Adapt	No adapt	Adapt
CNN+GRU	0.467	0.489 ↑1.2%	0.434	0.461 ↑2.7%	0.426	0.438 ↑1.2%
BiGRU	0.441	0.464 ↑2.3%	0.458	0.478 ↑2.0%	0.443	0.452 ↑0.9%
RNN	0.460	0.472 ↑1.2%	0.468	0.487 ↑1.9%	0.457	0.462 ↑0.5%
BiLSTM+Attn	0.472	0.553 ↑**8.1%**	0.470	0.481 ↑1.1%	0.443	0.458 ↑1.5%
cBERT	0.541	0.554 ↑1.3%	0.487	0.529 ↑4.2%	0.458	0.494 ↑3.6%
BERT	0.547	**0.581** ↑3.4%	0.501	0.525 ↑2.4%	0.454	0.514 ↑**6.0%**
fBERT	0.544	0.560 ↑1.6%	0.517	0.538 ↑2.1%	0.458	0.512 ↑5.4%
HateBERT	**0.557**	0.563 ↑0.6%	**0.522**	**0.545** ↑2.3%	0.480	**0.520** ↑4.0%

Table 3: Comparisons of performance (macro F1 scores) with and without Tad in out-domain settings averaged over ten epochs. "No adapt" retains the same modules as the models in in-domain experiments, whereas the "Adapt" incorporates Tad. The λ of knowledge infusion is set to 0.2. The highest scores in each column are in bold.

- CNN+GRU (Zhang, Robinson, and Tepper 2018): The model includes convolutional and max pooling layers to capture local textual features. We employ convolution filter sizes of 2, 3, and 4, each with 100 filters. Then, we max pool the outputs, passing the final vectors into a Gated Recurrent Unit layer and then fed into a dense layer for prediction.
- BiGRU (Ding, Zhou, and Zhang 2019): This method consists of a stack of bidirectional GRU and capsule network layers in its deep learning model.
- RNN (Mossie and Wang 2020): This method consists of passing BERT-encoded embeddings through two fully connected RNN layers.
- BiLSTM+Attn (Mathew et al. 2021): This model adds an attention layer after sequential layers of a BiLSTM model (Schuster and Paliwal 1997) and integrate attention weights into the final sentence representation.
- cBERT (Chiril et al. 2022): The pooled hidden states are fed into separate output layers for predicting hate speech and domain, respectively, and their loss terms are summed.
- BERT (Devlin et al. 2019): A transformer-based model is a stack of encoder layers with twelve fully connected neural networks augmented with self attention.
- fBERT (Sarkar et al. 2021) : This BERT model has undergone pretrained on an English offensive language corpus of millions of tweets.
- HateBERT (Caselli et al. 2021): An alternative BERT model pretrained on a dataset from banned Reddit communities compromising offensive, abusive, and hateful content.

For CNN+GRU and RNN-based models, we compute the hidden states by passing BERT-encoded embeddings through fully connected layers. For transformer-based models, we use the corresponding text representations by using Huggingface[3]. In fBERT and HateBERT, we replace BERT embeddings with the respective embeddings but keep the original modules of BERT. Max length of texts and batch size are set to 256 and 64. The learning rate is tuned in the range $[1e-6, 1e-4]$ on the validation set for optimal performance, and we report the best performance. The scheduler for the learning rate is applied via StepLR using PyTorch[4] with steps $= 8$ and $\gamma = 0.7$. The dimensions of the feed-forward hidden layers of all models are set to 256. We use dropout at different levels to regularize the outputs, where the drop probability is set to 0.5. The AdamW optimizer (Loshchilov and Hutter 2019) is used for transformer-based models with $\epsilon = 1e-8$ and RMSprop (Tieleman and Hinton 2012) is used for RNN-based models with $\gamma = 0.9$.

Model	Ethnicity	Religion	Sexorient
CNN+GRU	0.624	0.495	0.500
BiGRU	0.638	0.476	0.492
RNN	0.630	0.489	0.554
BiLSTM+Attn	0.640	0.559	0.518
cBERT	0.649	0.584	0.564
BERT	0.653	**0.635**	0.566
fBERT	0.672	0.596	0.584
HateBERT	**0.694**	0.616	**0.607**

Table 4: In-domain performance (macro F1) comparisons. The best scores of each column are in bold.

7 Results and Discussion

This section evaluates the selected models' performance in both in-domain and out-domain scenarios and then analyzes methods to address performance disparities.

7.1 Analysis: In-Domain and Out-Domain Performance Gaps

Table 3 and Table 4 report out-domain and in-domain performances over ten epochs, respectively. The results show that Tad is effective in adapting to domain shifts, resulting in reducing performance gaps across domains for all the models. We observe an increase of up to 8.1% in *Ethnicity* as shown

[3]https://huggingface.co/

[4]https://pytorch.org/docs/stable/optim.html

in Table 3. Notably, our methods exhibit the largest performance improvement for *Sexorient*, averaging 2.8% across all the eight experiments. Moreover, we find that models trained and tested on *Ethnicity* outperform other domains. We may explain this by *Ethnicity*'s large share of "Hate" instances. Specifically, as depicted in Table 1, *Ethnicity* exhibits the highest proportion of "Hate" instances, and *Sexorient* has the lowest. Table 3 displays that transformer-based models outperform others, with a peak of 58.1% on *Ethnicity* and a low of 45.4% on *Sexorient*.

7.2 Analysis: Performance Gaps between In-Domain and Out-Domain Experiments

While incorporating Tad is effective in enhancing cross-main experiments for the selected models, the results reveal notable performance gaps, which reach as high as 8.7%. The gaps can be attributed to data quality and quantity (Sarwar and Murdock 2022), such as data imbalance caused by insufficient domain-specific language (Chiril et al. 2022; Ludwig et al. 2022). Therefore, we focus on three components: (1) efficiency of external knowledge, (2) sufficiency of tuning data, and (3) balance of training data.

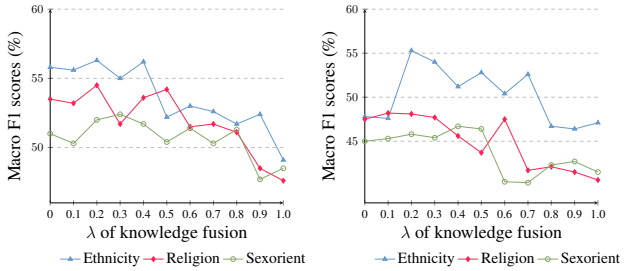
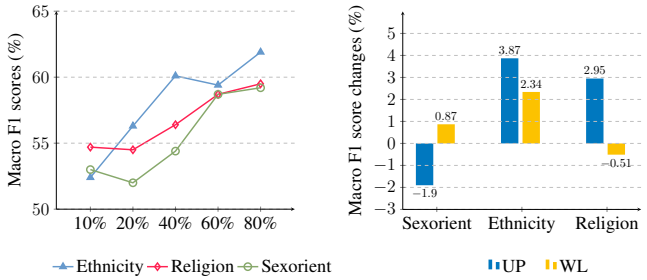


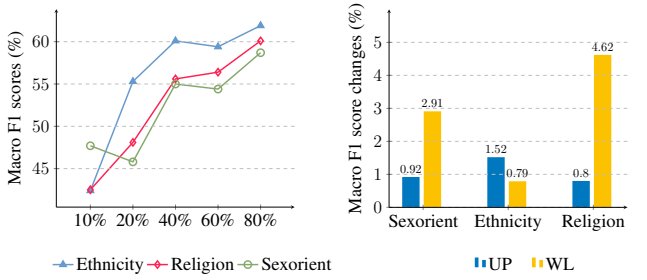
Figure 4: Performance changes for HateBERT+Tad and BiLSTM+Attn+Tad. Results for other models are omitted because they have similar trends. The results when $\lambda = 0.2$ match Table 3.

External Knowledge Concentration Affects Performance Figure 4 illustrates how performance changes due to the λ parameter of the knowledge infusion module. We observe a significant drop in performance when λ's value goes above 0.6. The result suggests that external knowledge may complicate model assessment, particularly when the prevalence of hateful tokens becomes excessive, ultimately leading to a decrease in performance. For instance, some words such as *idiot* (in HurtLex (Bassignana, Basile, and Patti 2018)) are prevalent on social media to express criticism or frustration, but do not necessarily constitute hate speech.

Tuning Data Size Affects Performance As shown in Figure 5, we report performance on randomly selected 10% to 80% of instances from each domain's full dataset to tune the weighting network. Increasing the data quantity generally improves performance across all three domains. This observation suggests that linguistic features of hate speech in real-world applications can be challenging to capture, especially when hateful words appear randomly.



(a) HateBERT+Tad.



(b) BiLSTM+Attn+Tad.

Figure 5: Data quality and quantity impacts. The left figures display the effect of development set sizes for tuning the weighting network. The right figures illustrate performance changes using UP and WL for balancing data distributions.

Data Imbalance Alleviation Results Vary Our dataset is heavily imbalanced, which can impair a model's performance. We propose two methods to alleviate the effect of imbalance:

Upsampling (UP): We upsample "Normal" instances with duplication to the same amount of "Hate" instances,

Weighted loss (WL): We upweight the underpopulated labels when computing the classification loss.

Figure 5 depicts the performance changes for HateBERT+Tad, with other models exhibiting similar trends and thus not included for brevity. We observe that the performance of WL and UP varying across models. Additionally, the WL approach demonstrates an improvement in the performance of up to 4.62% as shown in Figure 5, which suggests that data imbalance may be a significant challenge in hate speech detection (Sarwar and Murdock 2022). These observations corroborate previous results (Ludwig et al. 2022), which may be attributed to the complex landscape of hate speech detection.

7.3 Ablation Studies

To verify the contributions of each part of our models, we examine the contributions of knowledge infusion, domain-specific network, and weighting network, separately in out-domain experiments. Table 7 compares and reports the performance changes of HateBERT+Tad, where we use attention weights during prediction as token importance attributions (Jain et al. 2020). The table shows that HateBERT's performance can be improved up to 2.7% (*Sexorient*) by

Target	Hate Speech		Offensive		Normal	
	-	+	-	+	-	+
Ethnicity	0.657	0.705	0.406	0.433	0.462	0.482
Religion	0.613	0.655	0.230	0.326	0.541	0.562
Sexorient	0.464	0.485	0.382	0.483	0.532	0.546
$\Delta \uparrow (\%)$		3.1		7.5		1.5

Table 5: Averaged performance changes (macro F1 scores) of HateBERT+Tad in Table 3 over ten epochs of predicting the three labels. "+" represents with Tad and "-" represents without Tad.

adapting Tad. We find that incorporating a weighting network and domain descriptors generally performs better than knowledge infusion. Additionally, we note that the removal of the domain-specific network results in the most decrease in performance, reaffirming the value of learning domain-specific knowledge (Chiril et al. 2022; Ludwig et al. 2022). Moreover, Table 5 reports the performance changes of HateBERT with and without Tad for each label. We observe that our methods have the most positive impact on the "Offensive" prediction, and the least impact is observed on the "Normal" prediction.

7.4 Qualitative Analysis

We report the most common errors of HateBERT+Tad. We randomly select 100 samples and summarize the most common reasons for mispredictions, as shown in Table 6, along with examples.

Ambiguous usage Overuse of hateful words may lead the model to rely too much on these words, despite their potentially confounding nature.

Lack of slang The models struggle to identify content targeting specific communities through the use of slang and acronyms.

Words such as *ni**a* are prevalent on social media and are commonly used nonhatefully (Example (d)) and (Example (e)), which mislead the models to identify such contents as "Hate." Overused hate words such as in Example (a) may confound the models, especially when infusing external knowledge. Slang and jargon (e.g., acronyms) are widely used on social media; they introduce noise and degrade automated classification performance (Naseem, Razzak, and Eklund 2021), as seen in Example (c), highlighting the impact of negative knowledge shifts. Furthermore, when working with out-domain scenarios, the lack of domain-specific knowledge in the source domain can lead to performance degradation, making it challenging to generate domain-invariant representations, as seen in Examples (b) and (c). These examples demonstrate the complexity and nuances of hate speech in real-world applications.

8 Conclusion

Our paper introduces novel methods that use domain-aware networks to examine how performance varies across different targets in hate speech detection. Although our methods improve hate speech detection performance in cross-domain scenarios, there remain performance gaps between in-domain and out-domain settings, revealing challenges in achieving generalization in hate speech detection. We explore factors like data quantity, quality, and model structures to address performance disparities through extensive experiments. Our results reveal that additional hateful knowledge is crucial for broader applicability. Moreover, when incorporating domain adaptation, the most substantial performance improvement in out-domain experiments is observed for targets that frequently experience hate speech intertwined with other targets. Our findings provide valuable insights for improving hate speech detection and tackling the crucial challenge of domain shifts in hate language when it pertains to various target groups.

8.1 Limitations and Future Work

We acknowledge limitations in our approach. First, some source domains lack sufficient data, potentially leading to inconsistent results due to inadequate class label coverage. While HateXplain remains unique by including non-hateful instances annotated with target labels, there's potential for further exploration with other datasets. Our experiments are confined to a single dataset, making it challenging to generalize the main outcomes. Future work should explore experiments on other datasets, such as tweets (Zampieri et al. 2023). We acknowledge that while there are numerous advanced architectures for hate speech, offensive language, and abusive content detection models, detailed comparisons between these models are beyond the scope of this current work. We leave these comparative analyses for future research endeavors, such as unsupervised methods (Ahmad, Sujeeth, and Ekbal 2023), to address these limitations by incorporating a greater amount and higher diversity of data.

References

Ahmad, Z.; Sujeeth, V. S.; and Ekbal, A. 2023. Zero-Shot Hate to Non-Hate Text Conversion Using Lexical Constraints. *IEEE Transactions on Computational Social Systems*, 10(5): 2479–2488.

Almohaimeed, S.; Almohaimeed, S.; Shafin, A. A.; Carbunar, B.; and Bölöni, L. 2023. THOS: A Benchmark Dataset for Targeted Hate and Offensive Speech. *arXiv preprint arXiv:2311.06446*.

Arango, A.; Pérez, J.; and Poblete, B. 2019. Hate Speech Detection is Not as Easy as You May Think: A Closer Look at Model Validation. In *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval*, 45–54. New York: Association for Computing Machinery. ISBN 9781450361729.

Badjatiya, P.; Gupta, S.; Gupta, M.; and Varma, V. 2017. Deep Learning for Hate Speech Detection in Tweets. In *Proceedings of the International Conference on World Wide Web Companion*, 759–760. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee.

	Data	Error	Domain	Text	Prediction
(a)	♣	Ambiguous Usage	Sexorient	not pacsun though because im not a dyke	Hate
(b)	♦	Ambiguous Usage	Sexorient	i always wondered why weidner acted like a fa**ot	Normal
(c)	♦	Lack of Slang	Religion	singleshadeofislam to them women are less than things	Normal
(d)	♣	Ambiguous Usage	Ethnicity	i love when a ni**a say preciate it baby me	Hate
(e)	♦	Ambiguous Usage	Ethnicity	blow a stack for yo ni**as with yo trapping ass	Hate

Table 6: Randomly selected and summarized examples of incorrectly predicted examples of HateBERT+Tad, where the real labels for the instances are the opposite of those predicted. The ♦ and ♣ symbols indicate in-domain and out-domain settings. Tokens in ■ are the top three important attributions (ordered by attention weights obtained during training); darker color shades indicate greater strength.

Model	Ethnicity	Religion	Sexorient
HateBERT+Lex	0.558	0.535	0.510
HateBERT+W	0.561	0.541	0.514
HateBERT+DS	0.559	0.543	0.492
$\Delta\uparrow$ (%)	0.7	1.8	2.7
HateBERT+Tad-Lex	0.561	0.544	0.485
HateBERT+Tad-W	0.562	0.534	0.512
HateBERT+Tad-DS	0.558	0.525	0.511
$\Delta\downarrow$ (%)	0.3	1.4	1.7

Table 7: Out-domain performance for HateBERT+Tad (macro F1 scores), where other models exhibiting consistent trends. "+" represents with a module and "-" represents without a module. Here, $\Delta\uparrow$ and $\Delta\downarrow$ indicate the average performance increase and decrease over the results in Table 3.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*, abs/1409.0473.

Basile, V.; Bosco, C.; Fersini, E.; Nozza, D.; Patti, V.; Rangel Pardo, F. M.; Rosso, P.; and Sanguinetti, M. 2019. SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter. In *Proceedings of the International Workshop on Semantic Evaluation*, 54–63. Minneapolis: Association for Computational Linguistics.

Bassignana, E.; Basile, V.; and Patti, V. 2018. HurtLex: A multilingual lexicon of words to hurt. In *Italian Conference on Computational Linguistics (CLiC-it)*, volume 2253, 1–6. CEUR-WS, Torino: CEUR-WS.

Bose, T.; Illina, I.; and Fohr, D. 2021. Unsupervised Domain Adaptation in Cross-corpora Abusive Language Detection. In *Proceedings of International Workshop on Natural Language Processing for Social Media*, 113–122. Online: Association for Computational Linguistics.

Caselli, T.; Basile, V.; Mitrović, J.; and Granitzer, M. 2021. HateBERT: Retraining BERT for Abusive Language Detection in English. In *Proceedings of the Workshop on Online Abuse and Harms*, 17–25. Online: Association for Computational Linguistics.

Chiril, P.; Pamungkas, E. W.; Benamara, F.; Moriceau, V.; and Patti, V. 2022. Emotionally informed hate speech detection: A multi-target perspective. *Cognitive Computation*, 14: 1–31.

Davani, A. M.; Díaz, M.; and Prabhakaran, V. 2022. Dealing with Disagreements: Looking Beyond the Majority Vote in Subjective Annotations. *Transactions of the Association for Computational Linguistics*, 10: 92–110.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 4171–4186. Minneapolis: Association for Computational Linguistics.

Ding, Y.; Zhou, X.; and Zhang, X. 2019. YNU_DYX at SemEval-2019 task 5: A stacked BiGRU model based on capsule network in detection of hate. In *Proceedings of the International Workshop on Wemantic Evaluation*, 535–539.

Dixon, L.; Li, J.; Sorensen, J.; Thain, N.; and Vasserman, L. 2018. Measuring and Mitigating Unintended Bias in Text Classification. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, 67–73. New York: Association for Computing Machinery. ISBN 9781450360128.

ElSherief, M.; Nilizadeh, S.; Nguyen, D.; Vigna, G.; and Belding, E. 2018. Peer to Peer Hate: Hate speech instigators and their targets. In *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*, 1. California: AAAI press.

Fortuna, P.; Soler-Company, J.; and Wanner, L. 2021. How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets? *Information Processing and Management*, 58(3): 102524.

Ghosh, S.; Ekbal, A.; Bhattacharyya, P.; Saha, T.; Kumar, A.; and Srivastava, S. 2023. SEHC: A Benchmark Setup to Identify Online Hate Speech in English. *IEEE Transactions on Computational Social Systems*, 10(2): 760–770.

Huang, X.; Wormley, A.; and Cohen, A. 2022. Learning to Adapt Domain Shifts of Moral Values via Instance Weighting. In *Proceedings of ACM Conference on Hypertext and Social Media*, 121–131. Barcelona, Spain: Association for Computing Machinery.

Jain, S.; Wiegreffe, S.; Pinter, Y.; and Wallace, B. C. 2020. Learning to Faithfully Rationalize by Construction. In *Proceedings of the Annual Meeting of the Association for Com-*

putational Linguistics (ACL), 4459–4473. Online: Association for Computational Linguistics.

Jiang, A.; and Zubiaga, A. 2021. Cross-Lingual Capsule Network for Hate Speech Detection in Social Media. In *Proceedings of ACM Conference on Hypertext and Social Media*, 217–223. Online: Association for Computing Machinery.

Jiang, J.; and Zhai, C. 2007. Instance Weighting for Domain Adaptation in NLP. In *Proceedings of the Association of Computational Linguistics*, 264–271. Czech Republic: Association for Computational Linguistics.

Kamal, A.; Anwar, T.; Sejwal, V. K.; and Fazil, M. 2023. BiCapsHate: Attention to the Linguistic Context of Hate via Bidirectional Capsules and Hatebase. *IEEE Transactions on Computational Social Systems*, 1–12.

Liu, Q.; Zhang, Y.; and Liu, J. 2018. Learning domain representation for multi-domain sentiment classification. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 541–550. New Orleans: Association for Computational Linguistics.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*. New Orleans: OpenReview.net.

Ludwig, F.; Dolos, K.; Zesch, T.; and Hobley, E. 2022. Improving Generalization of Hate Speech Detection Systems to Novel Target Groups via Domain Adaptation. In *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*, 29–39. Online: Association for Computational Linguistics.

Lundberg, S. M.; and Lee, S.-I. 2017. A Unified Approach to Interpreting Model Predictions. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*, 4765–4774. Curran Associates, Inc.

Maity, K.; Sen, T.; Saha, S.; and Bhattacharyya, P. 2022. MTBullyGNN: A Graph Neural Network-Based Multitask Framework for Cyberbullying Detection. *IEEE Transactions on Computational Social Systems*, 1–10.

Mathew, B.; Dutt, R.; Goyal, P.; and Mukherjee, A. 2019. Spread of Hate Speech in Online Social Media. In *Proceedings of the ACM Conference on Web Science*, 173–182. New York: Association for Computing Machinery.

Mathew, B.; Saha, P.; Yimam, S. M.; Biemann, C.; Goyal, P.; and Mukherjee, A. 2021. HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection. In *Proceedings of AAAI Conference on Artificial Intelligence*, volume 35, 14867–14875. Online: AAAI Press.

Mossie, Z.; and Wang, J.-H. 2020. Vulnerable Community Identification Using Hate Speech Detection on Social Media. *Information Processing and Management*, 57(3): 102087.

Naseem, U.; Razzak, I.; and Eklund, P. W. 2021. A survey of Pre-Processing Techniques to Improve Short-Text Quality: A Case Study on Hate Speech Detection on Twitter. *Multimedia Tools and Applications*, 80: 35239–35266.

Ousidhoum, N.; Lin, Z.; Zhang, H.; Song, Y.; and Yeung, D.-Y. 2019. Multilingual and Multi-Aspect Hate Speech Analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, 4675–4684. Hong Kong, China: Association for Computational Linguistics.

Roy, P. K.; Bhawal, S.; and Subalalitha, C. N. 2022. Hate speech and offensive language detection in Dravidian languages using deep ensemble framework. *Computer Speech & Language*, 75: 101386.

Sarkar, D.; Zampieri, M.; Ranasinghe, T.; and Ororbia, A. 2021. fBERT: A Neural Transformer for Identifying Offensive Content. In *Findings of the Association for Computational Linguistics: EMNLP*, 1792–1798. Punta Cana, Dominican Republic: Association for Computational Linguistics.

Sarwar, S. M.; and Murdock, V. 2022. Unsupervised Domain Adaptation for Hate Speech Detection Using a Data Augmentation Approach. In *Proceedings of the International AAAI Conference on Web and Social Media*, 1, 852–862. Atlanta.

Schuster, M.; and Paliwal, K. K. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11): 2673–2681.

Swamy, S. D.; Jamatia, A.; and Gambäck, B. 2019. Studying Generalisability across Abusive Language Detection Datasets. In *Proceedings of Conference on Computational Natural Language Learning*, 940–950. Hong Kong, China: Association for Computational Linguistics.

Tieleman, T.; and Hinton, G. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2): 26–31.

Toraman, C.; Şahinuç, F.; and Yilmaz, E. 2022. Large-Scale Hate Speech Detection with Cross-Domain Transfer. In Calzolari, N.; Béchet, F.; Blache, P.; Choukri, K.; Cieri, C.; Declerck, T.; Goggi, S.; Isahara, H.; Maegaard, B.; Mariani, J.; Mazo, H.; Odijk, J.; and Piperidis, S., eds., *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 2215–2225. Marseille, France: European Language Resources Association.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. CA: Curran Associates, Inc.

Vigna, F. D.; Cimino, A.; Dell'Orletta, F.; Petrocchi, M.; and Tesconi, M. 2017. Hate Me, Hate Me Not: Hate Speech Detection on Facebook. In Armando, A.; Baldoni, R.; and Focardi, R., eds., *Proceedings of Italian Conference on Cybersecurity (ITASEC)*, 86–95. Venice, Italy: CEUR-WS.org.

Waseem, Z.; and Hovy, D. 2016. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In *Proceedings of the NAACL Student Research*

Workshop, 88–93. San Diego: Association for Computational Linguistics.

Wiegand, M.; Ruppenhofer, J.; and Kleinbauer, T. 2019. Detection of Abusive Language: The Problem of Biased Datasets. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 602–608. Minneapolis: Association for Computational Linguistics.

Zampieri, M.; Morgan, S.; North, K.; Ranasinghe, T.; Simmmons, A.; Khandelwal, P.; Rosenthal, S.; and Nakov, P. 2023. Target-Based Offensive Language Identification. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 762–770. Toronto, Canada: Association for Computational Linguistics.

Zhang, Z.; Robinson, D.; and Tepper, J. 2018. Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In Gangemi, A.; Navigli, R.; Vidal, M.-E.; Hitzler, P.; Troncy, R.; Hollink, L.; Tordai, A.; and Alam, M., eds., *The Semantic Web*, 745–760. Cham: Springer International Publishing.

Zhou, X.; Sap, M.; Swayamdipta, S.; Choi, Y.; and Smith, N. 2021. Challenges in Automated Debiasing for Toxic Language Detection. In *Proceedings of Conference of the European Chapter of the Association for Computational Linguistics*, 3143–3155. Online: Association for Computational Linguistics.

9 Paper Checklist

1. For most authors...
   (a) Would answering this research question advance science without violating social contracts, such as violating privacy norms, perpetuating unfair profiling, exacerbating the socio-economic divide, or implying disrespect to societies or cultures? Yes.
   (b) Do your main claims in the abstract and introduction accurately reflect the paper's contributions and scope? Yes.
   (c) Do you clarify how the proposed methodological approach is appropriate for the claims made? Yes.
   (d) Do you clarify what are possible artifacts in the data used, given population-specific distributions? No, our data is curated from previous work.
   (e) Did you describe the limitations of your work? Yes.
   (f) Did you discuss any potential negative societal impacts of your work? No, our framework has no potential negative societal impacts as far as we know.
   (g) Did you discuss any potential misuse of your work? No, our framework has no potential misuse as far as we know.
   (h) Did you describe steps taken to prevent or mitigate potential negative outcomes of the research, such as data and model documentation, data anonymization, responsible release, access control, and the reproducibility of findings? NA.
   (i) Have you read the ethics review guidelines and ensured that your paper conforms to them? Yes.
2. Additionally, if you ran machine learning experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? Yes.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? Yes.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? No.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? No.
   (e) Do you justify how the proposed evaluation is sufficient and appropriate to the claims made? Yes.
   (f) Do you discuss what is "the cost" of misclassification and fault (in)tolerance? Yes.
3. Additionally, if you are including theoretical proofs...
   (a) Did you state the full set of assumptions of all theoretical results? NA
   (b) Did you include complete proofs of all theoretical results? NA
4. Additionally, if you ran machine learning experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? We will release them if the paper is published.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? Yes.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? NA.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? Yes.
   (e) Do you justify how the proposed evaluation is sufficient and appropriate to the claims made? Yes.
   (f) Do you discuss what is "the cost" of misclassification and fault (in)tolerance? NA
5. Additionally, if you are using existing assets (e.g., code, data, models) or curating/releasing new assets, **without compromising anonymity**...
   (a) If your work uses existing assets, did you cite the creators? NA.
   (b) Did you mention the license of the assets? NA
   (c) Did you include any new assets in the supplemental material or as a URL? NA
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? No, the dataset is public.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? No, the data has no persoanl identifications.
   (f) If you are curating or releasing new datasets, did you discuss how you intend to make your datasets FAIR? NA.
   (g) If you are curating or releasing new datasets, did you create a Datasheet for the Dataset? NA.
6. Additionally, if you used crowdsourcing or conducted research with human subjects, **without compromising anonymity**...
   (a) Did you include the full text of instructions given to participants and screenshots? NA.
   (b) Did you describe any potential participant risks, with mentions of Institutional Review Board (IRB) approvals? NA.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? NA.
   (d) Did you discuss how data is stored, shared, and deidentified? NA.